**Documentul de specificare a cerin**

**Software Requirements Specification**

**(SRS) Document**

**Sistem de control termic al unei clădiri în scopul minimizării consumului energetic**

**14.04.2024**

**1.0**

**Andrei**

**Cuprins**

[1. Introducere 2](#_Toc164005367)

[1.1 Scopul 2](#_Toc164005368)

[1.2 Audiență țintă 2](#_Toc164005369)

[1.3 Sfera de aplicare 3](#_Toc164005370)

[2 Descriere generală 3](#_Toc164005371)

[2.1 Perspectiva produsului 3](#_Toc164005372)

[2.2 Caracteristici ale produsului 3](#_Toc164005373)

[2.3 Clase și caracteristici ale utilizatorilor 3](#_Toc164005374)

[2.4 Mediul de operare 4](#_Toc164005375)

[2.5 Constrângeri de proiectare și de implementare 4](#_Toc164005376)

[3 Cerințele sistemului 4](#_Toc164005377)

[3.1 Monitorizarea si preluarea parametrilor de mediu 4](#_Toc164005378)

[3.1.1 Descriere și Prioritate 4](#_Toc164005379)

[3.1.2 Secvențe Stimul/Răspuns 4](#_Toc164005380)

[3.1.3 Cerințe Funcționale 5](#_Toc164005381)

[3.2 *Reglarea parametrilor de temperatură din încăpere* 5](#_Toc164005382)

[3.2.1 Descriere și Prioritate 5](#_Toc164005383)

[3.2.2 Secvențe Stimul/Răspuns 5](#_Toc164005384)

[3.2.3 Cerințe Funcționale 6](#_Toc164005385)

[3.3 Stocarea datelor în cadrul unei baze de date 6](#_Toc164005386)

[3.3.1 Descriere și Prioritate 6](#_Toc164005387)

[3.3.2 Secvențe Stimul/Răspuns 6](#_Toc164005388)

[3.3.3 Cerințe Funcționale 6](#_Toc164005389)

[4 Cerințe pentru interfețe externe 7](#_Toc164005390)

[4.1 Interfețe cu utilizatorul 7](#_Toc164005391)

[4.2 Interfețe hardware 9](#_Toc164005392)

[4.3 Interfețe de comunicare 9](#_Toc164005393)

[4.4 Interfețe software 10](#_Toc164005394)

[5 Cerințe non-funcționale 11](#_Toc164005395)

[5.1 Cerințe de performanță 11](#_Toc164005396)

[5.2 Atribute de calitate ale software-ului 11](#_Toc164005397)

[6 Anexe 12](#_Toc164005398)

[6.1 Anexa B: Modele de Analiză 12](#_Toc164005399)

# Introducere

## Scopul

Scopul proiectului este crearea unei soluții practice și eficiente prin care se va realiza monitorizare și controlul unui sistem termic al unei clădiri cu scopul minimizării consumului de energie electrică. Prin implementarea unui sistem de control automat și a unei interfețe de utilizator intuitive, se va optimiza funcționarea sistemului în timp real, reducând consumul de energie fără a compromite confortul termic al ocupanților clădirii.

Obiectivele pe care proiectul intenționează să le îndeplinească sunt următoarele:

* Proiectarea și dezvoltarea unei machete de referință în care să fie inclus sistemul de control
* Dezvoltarea arhitecturii de control automată printr-un algoritm de predicție
* Dezvoltarea unei aplicații de monitorizare și control al parametrilor

## Audiență țintă

Descrie care parte a documentului SRS este destinată fiecărui cititor. Include o listă a tuturor părților interesate ale proiectului, dezvoltatorilor, managerilor de proiect și testerilor pentru o mai bună claritate.

Distribuția informației destinată fiecărui cititor este următoarea:

Dezvoltatori:

* Cerințele sistemului (Capitolul 3) - Dezvoltatorii vor folosi aceste informații pentru a înțelege exact ce trebuie să facă produsul și care sunt funcționalitățile pe care trebuie să le implementeze.
* Cerințe pentru interfețe externe (Capitolul 4) - Dezvoltatorii trebuie să înțeleagă aceste interfețe pentru a asigura compatibilitatea și integrarea corectă a produsului cu alte aplicații sau sisteme.
* Cerințe non-funcționale (Capitolul 5) – Dezvoltatorii pot fi interesați de modul de proiectare și implementare al sistemului în conformitate cu anumite standarde
* Anexe (Capitolul 7) – Dezvoltatorii pot accesa aceste anexe pentru a obține clarificări suplimentare

Client:

* Descrierea generală oferă o prezentare de ansamblu a produsului, inclusiv a perspectivei acestuia și a caracteristicilor sale, fiind utilă pentru înțelegerea scopului și a obiectivelor produsului.
* Clientul ar fi interesat de cerințele sistemului pentru a întelege mai bine funcționalitățile și caracteristicile specifice produsului.
* Cerințele pentru interfețe externe artă modul în care produsul interacționează cu alte sisteme sau componente, astfel clientul poate fi interesat de modul de integrare în infrastructura existentă

## Sfera de aplicare

Software-ul dezvoltat pentru monitorizarea și controlul sistemului termic al clădirii contribuie la îmbunătățirea eficienței operaționale, reducerea costurilor energetice și îmbunătățirea confortului clienților. Astfel, afacerea beneficiază de economii semnificative pe termen lung, îmbunătățirea reputației și satisfacției clienților, precum și creșterea eficienței operaționale.

# Descriere generală

## Perspectiva produsului

În contextul actual, clădirile reprezintă o parte semnificativă din consumul total de energie, iar eficiența energetică devine din ce în ce mai importantă în contextul luptei împotriva schimbărilor climatice și al creșterii costurilor energetice. Astfel, există o presiune tot mai mare pentru dezvoltarea și implementarea soluțiilor de control termic care să permită clădirilor să funcționeze în mod eficient din punct de vedere energetic, reducând impactul asupra mediului și costurile asociate cu consumul de energie electrică.

## Caracteristici ale produsului

Software-ul va permite afișarea în timp real a valorilor măsurate de către senzorii de temperatură și alți senzori relevanți. De asemenea datele legate de temperatură și umiditate sunt preluate la un interval de 10 minute, fiind un interval optim pentru detectarea schimbărilor semnificative în mediul înconjurător, dar și pentru econimisirea energiei. Aceste date sunt stocate într-o bază de date, ulterior fiind prelucrate.

Softul va include un algoritm care va analiza datele senzorilor și va regla automat sistemul de încălzire/răcire pentru a menține temperatura interioară la nivelul dorit, în funcție de intervalele orare definite de utilizator.

Utilizatorii vor avea posibilitatea de a interveni manual pentru a regla temperatura în clădire, dacă este necesar, prin intermediul unei interfețe intuitive.

## Clase și caracteristici ale utilizatorilor

Software-ul este destinat atât birourilor firmelor, cât și locuințelor obișnuite, cu utilizatori care pot varia în funcție de mediu și nevoi.

În funcție de frecvența de utilizare, unii utilizatori pot interacționa cu software-ul în mod regulat, ca parte integrantă a activităților lor zilnice în birou sau acasă, în timp ce alții pot utiliza software-ul doar ocazional, în situații specifice sau pentru anumite sarcini.

În funcție de funcțiile utilizate, există utilizatori care se concentrează mai mult pe funcțiile de monitorizare și raportare a consumului de energie, fiind interesați în special de informațiile legate de eficiența energetică a clădirii, în timp ce alții pot folosi software-ul în principal pentru controlul și reglarea temperaturii în spațiul lor.

## Mediul de operare

Softul este proiectat sa ruleze pe o placa Raspberry Pi 4B care are 4GB de RAM, o memorie totală de stocare de 64 GB pe un card SD și un sistem de operare Raspberry Pi OS, bazat pe Linux.

Codul sursă dezvoltat pentru această aplicație a fost realizat în Python, în cadrul căruia au fost importat mai multe librării utilizate pentru conectarea dispozitivelor utilizate.

## Constrângeri de proiectare și de implementare

Constrângerile de proiectare și implementare pentru acest proiect au fost variate, având atât aspecte legate de hardware, cât și de software.

Pe partea hardware, principala constrângere a fost reprezentată de caracteristicile și capacitățile plăcii Raspberry Pi 4B, inclusiv procesorul, memoria RAM și porturile de interfață disponibile. De asemenea, adaptarea la diverse tipuri de senzori și dispozitive externe a necesitat o atenție deosebită pentru a asigura compatibilitatea și funcționalitatea adecvată. Pe partea de software, una dintre constrângerile majore a fost reprezentată de utilizarea limbajului de programare Python, care, deși oferă o flexibilitate și o ușurință de dezvoltare considerabile, poate avea anumite limitări în ceea ce privește performanța și eficiența în comparație cu alte limbaje de programare.

În plus, integrarea și comunicarea cu alte aplicații sau servicii externe au implicat gestionarea protocoalelor de comunicare și a aspectelor legate de securitate, pentru a asigura o interacțiune corespunzătoare și sigură între diferitele componente ale sistemului.

# Cerințele sistemului

## Monitorizarea si preluarea parametrilor de mediu

### Descriere și Prioritate

Datele vor fi preluate prin intermediul senzorilor de calitate, dintre care se enumera cei de temperatura, umiditate, radiatie solara, fotorezistente. Pe langa acesti senzori, se va amplasa pe acoperisul constructiei o statie pluviometrica, care va prelua date precum viteza vantului, cantitatea de precipitatii, presiunea atmosferica si temperatura din aer. Această funcționalitate are o prioritate ridicată, fiind necesară pentru controlarea efectivă a sistemului.

### Secvențe Stimul/Răspuns

Senzorii preiau automat datele din mediul în care sunt poziționați, fără a interacționa cu utilizatorul, apoi sunt transmise în continuare către dispozitivele de automatizare, care le vor prelucra în conformitate cu un anumit set de reguli bine stabilite.

### Cerințe Funcționale

1. preluare informațiilor de mediu
2. transmiterea informațiilor de la senzori la sistemul de prelucrare
3. arhivarea informațiilor de mediu
4. prezentarea datelor de mediu în formă tabelară
5. prezentarea datelor de mediu în formă grafică
6. prezentarea de statistici pe baza datelor prelucrate: minim, maxim, valori medie zi, valori medie noapte

Funcționând într-un mod neintruziv, senzorii colectează date precise în timp real și le transmit în continuare către dispozitivele de automatizare. Aceste dispozitive, echipate cu algoritmi și reguli bine stabilite, prelucrează informațiile primite pentru a lua decizii și pentru a iniția acțiuni specifice. In cazul in care unul din senzori nu poate transmite valori, din cauza unei defecțiuni, se va afișa o eroare în acest sens.

## *Reglarea parametrilor de temperatură din încăpere*

### Descriere și Prioritate

Utilizatorul poate regla temperatură se va face în funcție de mai multe criterii. Unul dintre acestea ar fi intervalul orar în care este utilizat spațiul util destinat utilizării încăperii. În momentul în care încăperea nu este utilizată, temperatura va scădea, până la un anumit prag, pentru a evita un consum energetic crescut, astfel sporind eficiența sistemului. Un alt criteriu ar fi în funcție de preferințele utilizatorului, care poate modifica temperatura astfel încât acesta să se regăsească într-o stare de confort termic favorabilă.

### Secvențe Stimul/Răspuns

Temperatura se reglează automat în funcție de programele orare predefinite de către utilizator, care pot fi suprascrise, pe baza unui algoritm de predicție. Sistemul va anticipa timpul la care va trebui să pornească sistemul, înaintea orei de pornire, pentru a realizat reglarea treptat, iar în momentul în care se atinge ora stabilită, temperatura să se alfe deja la acel nivel stabilit.

Utilizatorul va avea posibilitatea de a regla temperatura în încăpere printr-un potențiometru amplasat în mod convenabil în interiorul acesteia. Acest dispozitiv simplu și eficient îi oferă utilizatorului controlul asupra mediului ambiental, permițându-i să ajusteze temperatura în funcție de preferințele personale și de condițiile de confort dorite.

### Cerințe Funcționale

REQ-1: Reglarea temperaturii în funcție de program orar

REQ-2: Sistemul ofera posibilitatea utilizatorului de a modifica temperatura în încăpere conform preferințelor personale

REQ-3: Software-ul realizeaza reglarea automată a parametrilor, prin predicție, eliminând necesitatea intervenției directe a utilizatorului.

REQ-4: Sistemul trebuie să fie proiectat pentru a anticipa și gestiona eventualele condiții de eroare.

## Stocarea datelor în cadrul unei baze de date

### Descriere și Prioritate

Datele preluate cu ajutorul senzorilor de calitate, care monitorizează constant parametri relevanți, vor fi înregistrate și stocate într-o bază de date centralizată. Acest depozit de informații oferă un mijloc eficient de colectare și organizare a datelor, facilitând accesul și gestionarea acestora. Ulterior, prin intermediul unui proces automatizat, informațiile stocate vor fi extrase din baza de date pentru a oferi utilizatorilor o perspectivă detaliată asupra calității și performanței, permițând luarea deciziilor informate și aplicarea de îmbunătățiri în funcție de nevoile identificate.

Baza de date este utilizată pentru a stoca o cantitate suficient de mare de date istorice pentru a facilita sistemul de predicție.

### Secvențe Stimul/Răspuns

Utilizatorul accesează interfața sistemului pentru a obține informații despre calitatea și performanța sistemului într-un anumit interval de timp, iar sistemul, în urma solicitării utilizatorului, inițiază procesul de extragere a datelor din baza de date centralizată, unde sunt stocate informații preluate de senzorii de calitate.

Utilizatorul primește o prezentare detaliată și organizată a datelor referitoare la parametri relevanți, cum ar fi calitatea și performanța sistemului. Sistemul furnizează informații clare și coerente, permițând utilizatorului să înțeleagă în mod eficient calitatea și performanța sistemului.

### Cerințe Funcționale

REQ-1: Sistemul trebuie să fie capabil să înregistreze și să stocheze datele colectate într-o bază de date centralizată, asigurând integritatea și securitatea acestora.

REQ-2: Software-ul ofera un mecanism eficient pentru colectarea și organizarea datelor în cadrul bazei de date, facilitând accesul și gestionarea acestora într-un mod coerent și ușor de înțeles.

REQ-3: Software-ul trebuie să fie proiectat pentru a anticipa și gestiona eventualele condiții de eroare în timpul colectării, înregistrării sau extragerii datelor, asigurând funcționarea corectă și stabilă a sistemului.

# Cerințe pentru interfețe externe

## Interfețe cu utilizatorul

Logica din spatele interacțiunilor dintre utilizatori și software. Aici se pot include ecranului demonstrative, butoanele și funcțiile care ar apărea pe fiecare ecran, mesajele care urmează să fie afișate pe fiecare ecran și ghidurile de stil care urmează să fie utilizate. Detaliile designului interfeței utilizatorului ar trebui să fie documentate într-o specificație separată a interfeței utilizatorului.

Utilizatorul dispune de un număr predefit de interfețe cu care poate interacționa.

În cadrul primei interfețe sunt afișate detalii legate de temperatura interioară, dar și un control care afișează ora curentă, acestea fiind singurele funcționalități ale acestei ferestre. Utilizatorul poate da click pe ecran pentru a comuta interfața într-un nou ecran.

A white rectangular sign with black numbers

Description automatically generated

Figura 1 - Primul ecran

Al doilea ecran al aplicației include un meniu principal, în care utilizatorul are posibilitatea de a alege dintre două variante, și anume să acceseze un panou de vizualizare sau un panou de control. Această interacțiune se realizează prin click pe ecran, pe unul din cele două butoane. Suplimentar, pe această interfață se regăsește numele ecranului “Meniu principal” și două controale în care sunt afișate detalii despre data și ora corectă.

A blue circles with white text

Description automatically generated

Figura 2 - Ecranul Meniu Principal

Dacă utilizatorul alege să acceseze ecranul de vizualizare al parametrilor, interfața se va schimbă ca în figura 3, de mai jos. În cadrul acestei interfețe pot fi vizualizate datele aferente fiecărui senzor, în timp real. Valorile din controale se schimbă în mod dinamic pe măsură ce senzorii transmit date, iar cele trei din partea dreaptă își modifică culoarea în funcție de valoarea măsurată, verde însemnând că este o valoare bună, iar roșu pentru o valoare negativă.

De asemenea, în bara de meniu de sus există un buton care permite revenirea la ecranul anterior, numele ecranului, și ora curentă.

A screenshot of a screen

Description automatically generated

Figura 3 - Ecranul de vizualizare al parametrilor

Dacă utilizatorul alege să acceseze ecranul de control al temperaturii, interfața se modifică, ca în imaginea de mai jos. În cadrul acestui ecran sunt mai multe controale cu care utilizatorul poate interacționa.

A screenshot of a computer

Description automatically generated

Figura 4 - Ecranul de control al temperaturii

## Interfețe hardware

Software-ul utilizează o gamă de dispozitive hardware pentru funcționarea sa. Acestea includ placa Raspberry Pi 4B ca platformă principală, împreună cu senzorii DHT22, MQ2, MQ4 pentru detectarea diverselor gaze, un senzor pentru particule fine PM 2.5, un senzor UV și un senzor cu fotorezistor. Pentru prototipare și conectarea dispozitivelor, se utilizează o placa breadboard.

Pentru transmiterea datelor către o bază de date, este necesară conectivitatea la internet, fie prin Wi-Fi, fie prin cablu Ethernet. În ceea ce privește protocoalele de comunicare, acestea pot include TCP/IP pentru transmiterea datelor în rețea și MQTT pentru comunicarea între dispozitive și server. Utilizarea acestor protocoale asigură o interacțiune eficientă și sigură între hardware și software.

## Interfețe de comunicare

Proiectul utilizează diverse standarde de comunicare pentru a facilita interacțiunea între diferitele componente ale sistemului:

I2C (Inter-Integrated Circuit):

* Folosit pentru conectarea unui display la Raspberry Pi.
* Permite transmiterea datelor între Raspberry Pi și display-ul conectat prin intermediul protocolului I2C.
* Acesta oferă o interfață simplă și eficientă pentru afișarea informațiilor pe display-ul conectat.

TCP/IP (Transmission Control Protocol/Internet Protocol):

* Folosit pentru comunicarea cu baza de date.
* Permite transferul datelor între Raspberry Pi și baza de date folosind protocolul TCP/IP.
* Acest lucru facilitează stocarea și accesul la date într-o bază de date centralizată, permițând monitorizarea și gestionarea eficientă a informațiilor colectate de sistem.

## Interfețe software

Frontend-ul și backend-ul sunt implementate în Python, folosindu-se această limbaj de programare pentru dezvoltarea ambelor componente. Interfața grafică este realizată cu biblioteca Tkinter, care permite crearea de interfețe grafice utilizator (GUI) în Python.

Pentru stocarea datelor, se utilizează Firebase Realtime Database, oferind o bază de date cloud în timp real pentru aplicație.

Sunt utilizate diverse biblioteci și framework-uri pentru funcționarea adecvată a aplicației, printre care:

* time: Oferă funcționalități legate de manipularea timpului, inclusiv obținerea timpului curent, gestionarea întârzierilor și a pauzelor în execuția programului.
* tkinter: Bibliotecă standard în Python pentru dezvoltarea interfeței grafice (GUI), oferind instrumente și elemente grafice pentru crearea ferestrelor, butoanelor, etichetelor etc.
* customtkinter: Presupun că este o bibliotecă personalizată, adaptată nevoilor specifice, care poate oferi extensii sau funcționalități adiționale pentru Tkinter.
* sys: Oferă acces la diferite variabile și funcționalități specifice sistemului, cum ar fi calea către interpretorul Python și argumentele liniei de comandă.
* PIL (Python Imaging Library): Pentru manipularea imaginilor în Python, inclusiv încărcarea, manipularea și salvarea imaginilor în diferite formate de fișiere.
* board: Parte din pachetul Adafruit CircuitPython, oferă suport pentru interfața cu placa Raspberry Pi și alte dispozitive hardware.
* adafruit\_dht: Bibliotecă pentru interacțiunea cu senzorul de temperatură și umiditate DHT, furnizând metode pentru citirea și interpretarea datelor de la acest senzor.
* pyrebase: Pentru interacțiunea cu serviciul Firebase, permițând comunicarea cu baza de date Firebase în Realtime Database.
* datetime: Oferă funcționalități pentru lucrul cu date și timp în Python, inclusiv obținerea datelor curente și manipularea datelor calendaristice.
* threading: Pentru lucrul cu fire de execuție (threads) în Python, permițând execuția de cod simultan și gestionarea concurenței.
* RPi.GPIO: Pentru interacțiunea cu pinii GPIO ai plăcii Raspberry Pi, permițând controlul și monitorizarea diferitelor dispozitive conectate la acești pini.
* matplotlib.pyplot: Pentru crearea de grafice și diagrame în Python, permițând vizualizarea datelor într-un mod grafic și interactiv.
* numpy: Pentru calcul numeric și manipularea matricelor și tabelelor de date în Python, oferind funcționalități avansate pentru analiza datelor.

# Cerințe non-funcționale

## Cerințe de performanță

În ceea ce privește gestionarea timpului în cadrul sistemului, există câteva aspecte importante de luat în considerare. Senzorii responsabili cu măsurarea temperaturii și umidității efectuează preluarea datelor o dată la fiecare 10 minute. Acest interval a fost selectat pentru a maximiza eficiența energetică și pentru a asigura o monitorizare adecvată a condițiilor de mediu în încăpere.

Pe de altă parte, celelalte tipuri de senzori, care monitorizează aspecte precum calitatea aerului și starea luminii, preiau date la intervale mai scurte, o dată la fiecare minut. Această frecvență mai mare de preluare a datelor este justificată de caracterul dinamic al informațiilor pe care acești senzori le monitorizează.

Datele preluate de către senzori sunt încărcate în baza de date a sistemului o dată la fiecare 10 minute. Această sincronizare a încărcării datelor în baza de date contribuie la organizarea eficientă a informațiilor și la asigurarea unei funcționări fluide a sistemului.

În ceea ce privește interacțiunea utilizatorului cu sistemul, acesta revine automat la interfața inițială dacă nu se efectuează nicio interacțiune într-un interval de 30 de secunde. Această funcționalitate optimizează experiența utilizatorului și contribuie la eficiența generală a sistemului.

## Atribute de calitate ale software-ului

Bazându-ne pe informațiile furnizate, codul software este însoțit de comentarii care facilitează înțelegerea și întreținerea acestuia. Aceste comentarii sunt esențiale pentru dezvoltatori, oferindu-le o mai bună înțelegere a funcționalității fiecărei părți a codului și permițându-le să facă modificări sau îmbunătățiri ulterioare într-un mod eficient.

De asemenea, software-ul este compatibil cu platforma Raspberry Pi 4B, ceea ce oferă utilizatorilor flexibilitatea de a-l utiliza pe această platformă specifică. Această compatibilitate poate fi un aspect important pentru utilizatorii care doresc să implementeze soluția pe hardware-ul specificat.

În plus, partea hardware a sistemului poate fi extinsă prin adăugarea de mai mulți senzori. Această extensibilitate permite utilizatorilor să-și personalizeze sistemul în funcție de nevoile lor specifice și să adauge funcționalități suplimentare în viitor fără a fi necesare modificări semnificative în codul software.

# Anexe

## Anexa B: Modele de Analiză

A diagram of a process flow

Description automatically generated

Figura 5 - Diagramă de stare

A diagram of a computer server

Description automatically generated

Figura 6 - Arhitectura proiectului